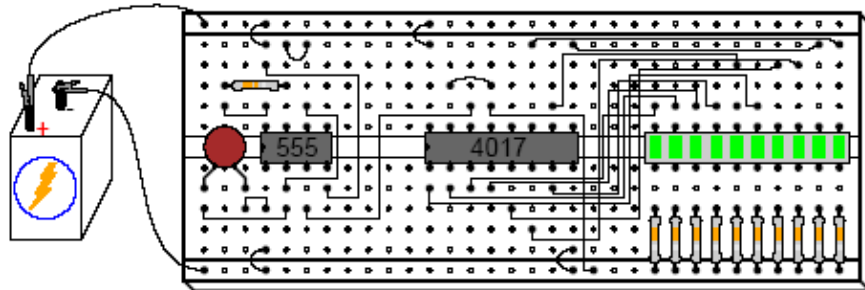


CÔNG LOGIC VÀ ĐẠI SỐ BOOLE

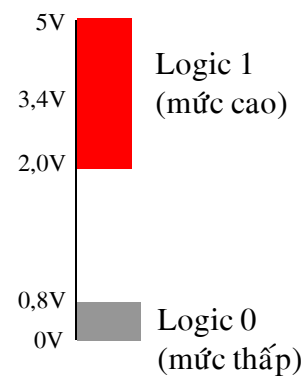


1

I. TRẠNG THÁI LOGIC 0 VÀ LOGIC 1

LOGIC 0	LOGIC 1
Sai	Đúng
Tắt	Mở
Thấp	Cao
Không	Đồng ý
Giả	Thật

Mức logic:



2

Số nhị phân có số
mã là 0,1 và cơ số
là 2

Ví dụ:

$$112_D = \frac{0111}{7} \frac{0000}{0}_B = 70_H$$

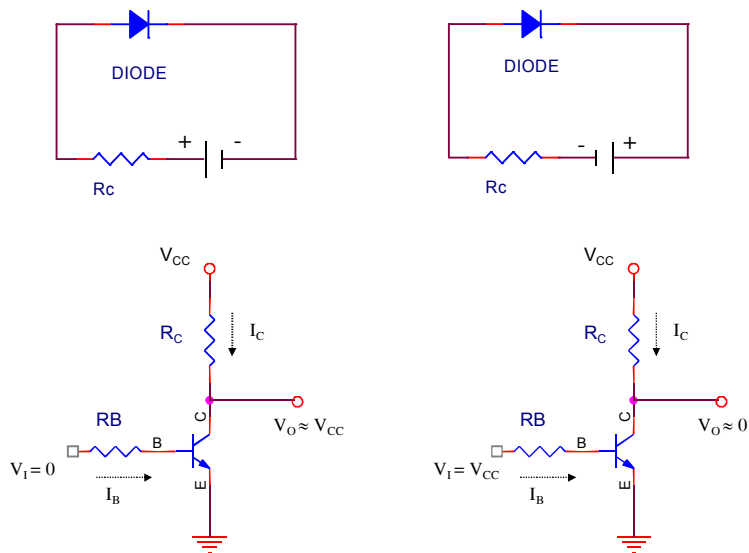
D: decimal

B: binary

H: hexadecimal

Số thập phân	Số thập lục	Số nhị phân
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

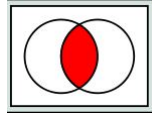
3



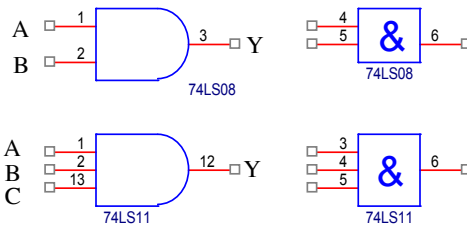
4

II. CÁC CỔNG (HÀM) LOGIC

1. CỔNG AND



Bảng trạng thái (bảng sự thật): tìm trạng thái ngõ ra theo điều kiện ngõ vào

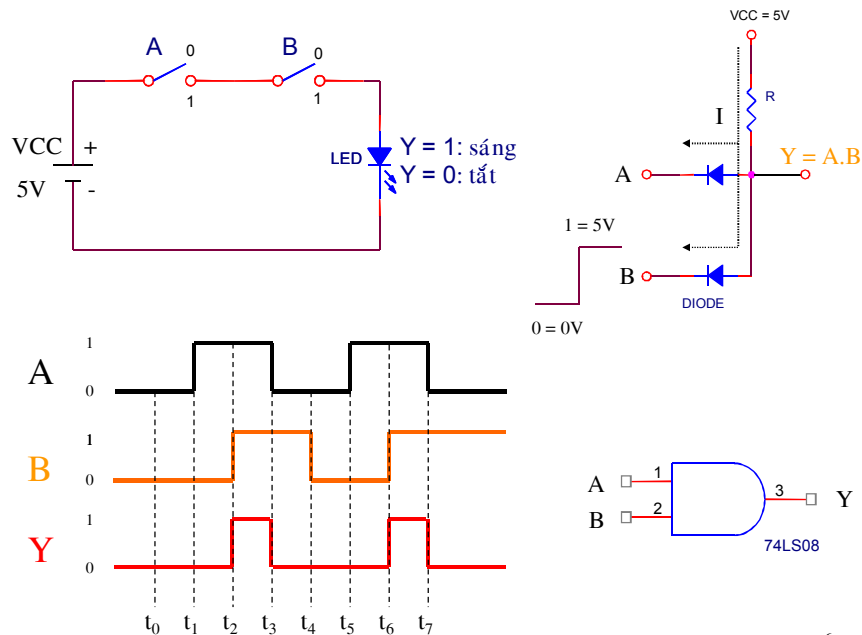


$$Y = A.B \quad (\text{đọc: } Y \text{ bằng } A \text{ VÀ } B)$$

- $A = 0 \rightarrow Y = 0$ bất chấp B
- $A = 1 \rightarrow Y = B$

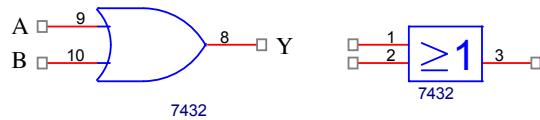
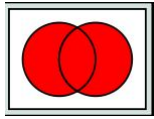
Biến số		Hàm số
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

5



6

2. CÔNG OR



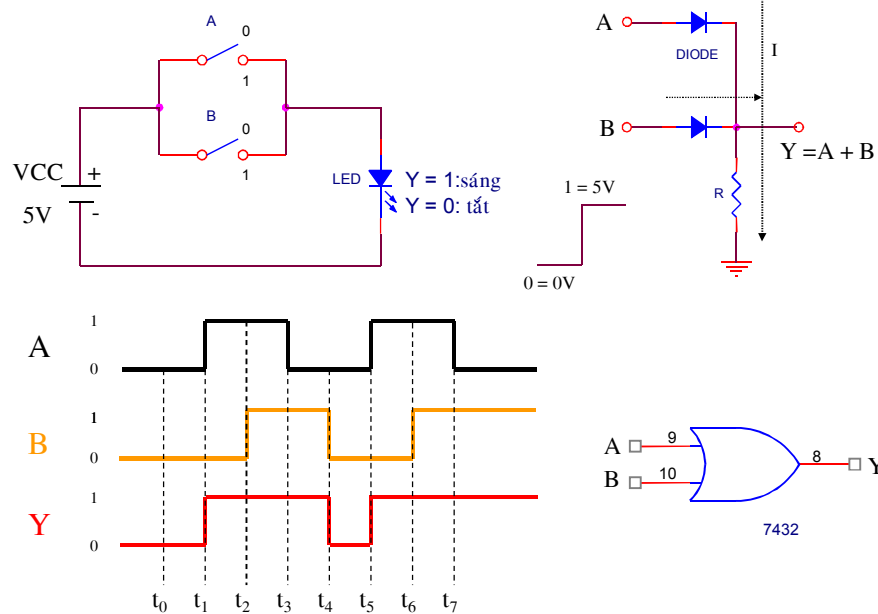
$$Y = A + B \quad (\text{đọc: } Y \text{ bằng } A \text{ HOẶC } B)$$

- $A = 0 \rightarrow Y = B$
- $A = 1 \rightarrow Y = 1$ bất chấp B

Bảng trạng thái:

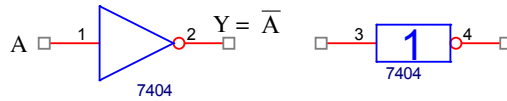
Biến số		Hàm số
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

7



8

3. CỘNG NOT



$$Y = \overline{A}$$

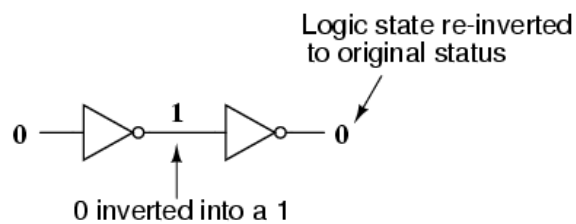
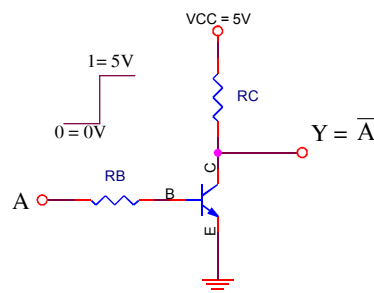
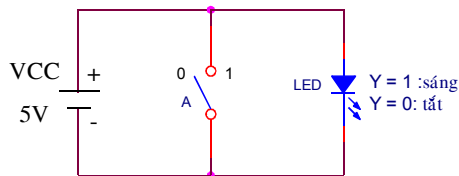
(đọc: Y bằng A KHÔNG B)

Bảng trạng thái:

Biến số	Hàm số
A	Y
0	1
1	0

- Chỉ có một ngõ vào và một ngõ ra

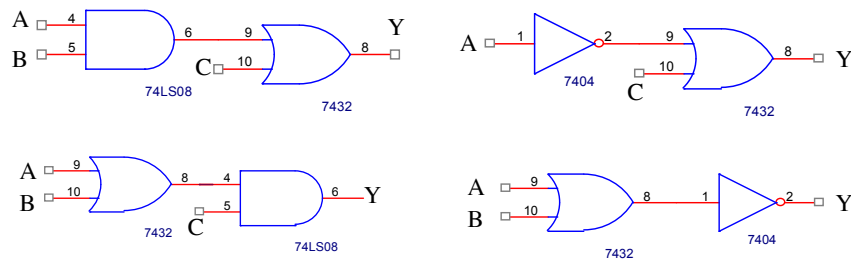
9



10

Tóm tắt

OR	AND	NOT
$0 + 0 = 0$ $0 + 1 = 1$ $1 + 0 = 1$ $1 + 1 = 1$	$0 \cdot 0 = 0$ $0 \cdot 1 = 0$ $1 \cdot 0 = 0$ $1 \cdot 1 = 1$	$\overline{0} = 1$ $\overline{1} = 0$

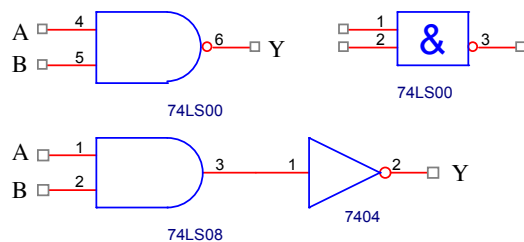


VD: a) Vẽ mạch $x = AB + \overline{B}C$

b) Vẽ mạch $y = AC + \overline{B}C + \overline{A}BC$

11

4. CÔNG NAND



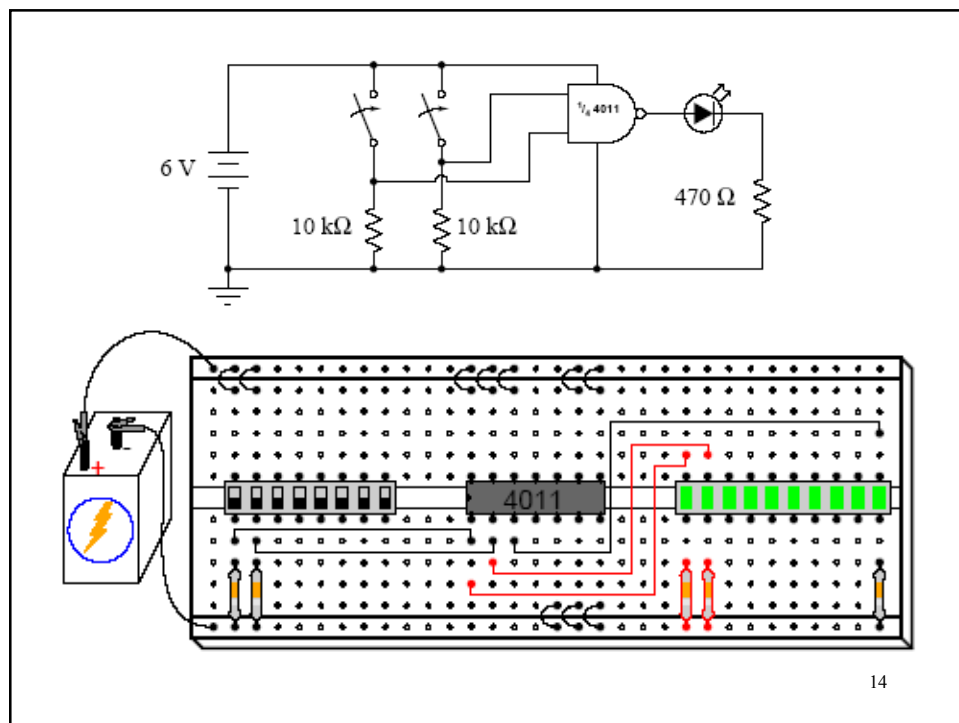
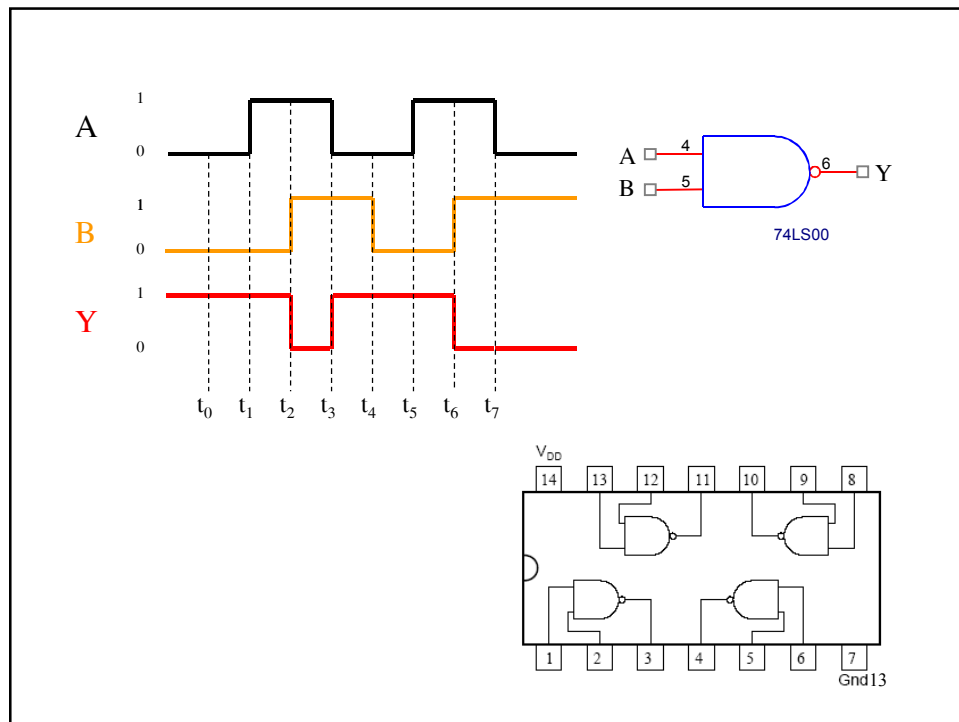
$$Y = \overline{AB}$$

- $A = 0 \rightarrow Y = 1$ bất chấp B
- $A = 1 \rightarrow Y = \overline{B}$

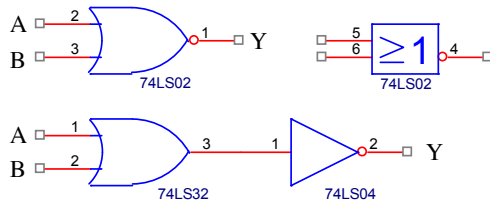
Bảng trạng thái:

Biến số		Hàm số
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

12



5. CÔNG NOR



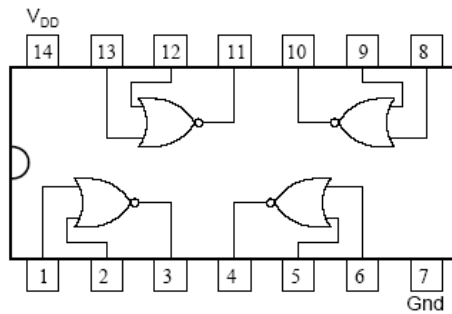
$$Y = \overline{A + B}$$

- A = 1 -> Y = 0
- A = 0 -> Y = B

Bảng trạng thái:

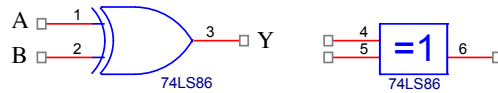
Biến số		Hàm số
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

15



16

6. CÔNG EX-OR (EXCLUSIVE-OR)



Bảng trạng thái:

Biến số		Hàm số
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$$Y = A \oplus B$$

$$= \overline{A}B + A\overline{B}$$

- Cùng trạng thái ngõ ra = 0
- Khác trạng thái ngõ ra = 1

17

III. ĐẠI SỐ BOOLE

- Mạch logic (mạch số) hoạt động dựa trên chế độ nhị phân:
 - Điện thế ở đầu vào, đầu vào hoặc bằng 0, hoặc bằng 1
 - Với 0 hay 1 tượng trưng cho các khoảng điện thế được định nghĩa sẵn
 - VD: $0 \rightarrow 0.8V : 0$
 $2.0 \rightarrow 5V : 1$
- ➡ Cho phép ta sử dụng Đại số Boole như là một công cụ để phân tích và thiết kế các hệ thống số

18

- **Đại số Boole:**

- Do George Boole sáng lập vào thế kỷ 19
- Các hằng, biến và hàm chỉ nhận 1 trong 2 giá trị: 0 và 1
- Là công cụ toán học khá đơn giản cho phép mô tả mối liên hệ giữa các đầu ra của mạch logic với các đầu vào của nó dưới dạng biểu thức logic
- Là cơ sở lý thuyết, là công cụ cho phép nghiên cứu, mô tả, phân tích, thiết kế và xây dựng các hệ thống số, hệ thống logic, mạch số ngày nay.

19

Các phép tính khi áp dụng cho logic 0 và logic 1:

OR	AND	NOT
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\overline{0} = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\overline{1} = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

Các định lý:

- | | |
|--------------------------------|----------------------------|
| (1) $X \cdot 0 = 0$ | (5) $X + 0 = X$ |
| (2) $X \cdot 1 = X$ | (6) $X + 1 = 1$ |
| (3) $X \cdot X = X$ | (7) $X + X = X$ |
| (4) $X \cdot \overline{X} = 0$ | (8) $X + \overline{X} = 1$ |

20

(9) $X + Y = Y + X$ (giao hoán)

(10) $X \cdot Y = Y \cdot X$ (giao hoán)

(11) $X + (Y + Z) = (X + Y) + Z = X + Y + Z$ (phối hợp)

(12) $X(YZ) = (XY)Z = XYZ$ (phối hợp)

(13a) $X(Y + Z) = XY + XZ$ (phân bố)

(13b) $(W + X)(Y + Z) = WY + XY + WZ + XZ$ (phân bố)

(14) $X + XY = X$

(15) $X + \overline{X}Y = X + Y$

21

* Định luật De Morgan:

(16) $\overline{X + Y} = \overline{X} \cdot \overline{Y}$

(17) $\overline{X \cdot Y} = \overline{X} + \overline{Y}$

* VD: đơn giản biểu thức $z = \overline{(\overline{A} + C) \cdot (B + \overline{D})}$

Giải:

$$\begin{aligned} z &= \overline{(\overline{A} + C) \cdot (B + \overline{D})} \\ &= \overline{(\overline{A} + C)} + \overline{(B + \overline{D})} \\ &= (\overline{\overline{A}} \cdot \overline{C}) + (\overline{B} \cdot \overline{\overline{D}}) \\ &= A\overline{C} + \overline{B}D \end{aligned}$$

22

* VD: thiết kế mạch chỉ sử dụng 1 cổng NAND và 1 cổng NOT có biểu thức như sau :

$$z = \overline{A} + \overline{B} + C$$

Giải:

$$\begin{aligned} z &= \overline{A} + \overline{B} + C \\ &= \overline{\overline{\overline{A} + \overline{B} + C}} \\ &= \overline{\overline{A} \cdot \overline{B} \cdot \overline{C}} \\ &= \overline{A \cdot B \cdot C} \end{aligned}$$

Hoặc

$$\begin{aligned} z &= \overline{A} + \overline{B} + C \\ &= \overline{A} + \overline{B} + \overline{\overline{C}} \\ &= \overline{A \cdot B \cdot C} \end{aligned}$$

23

• Hàm Boole (Boolean Function):

1. Định nghĩa:

* Hàm Boole là 1 biểu thức được tạo bởi các biến nhị phân và các phép toán nhị phân NOT, AND, OR.

$$F(x, y, z) = x \cdot y + \overline{x} \cdot \overline{y} \cdot z$$

* Với giá trị cho trước của các biến, hàm Boole sẽ có giá trị là 0 hoặc 1.

* Bảng giá trị:

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

24

2. Bù của 1 hàm:

- Sử dụng định luật De Morgan:

$$F = x \cdot y + \bar{x} \cdot \bar{y} \cdot z$$

$$\begin{aligned}\bar{F} &= \overline{x \cdot y + \bar{x} \cdot \bar{y} \cdot z} \\ &= (\overline{x \cdot y}) \cdot (\overline{\bar{x} \cdot \bar{y} \cdot z})\end{aligned}$$

$$\bar{F} = (\bar{x} + \bar{y}) \cdot (x + y + \bar{z})$$

- Lấy biểu thức đối ngẫu và lấy bù các biến:

- * **Tính đối ngẫu (Duality):** Hai biểu thức được gọi là đối ngẫu của nhau khi ta thay phép toán AND bằng OR, phép toán OR bằng AND, 0 thành 1 và 1 thành 0.

$$F = x \cdot y + \bar{x} \cdot \bar{y} \cdot z$$

$$\text{Lấy đối ngẫu: } F = (x + y) \cdot (\bar{x} + \bar{y} + z)$$

$$\text{Bù các biến: } \bar{F} = (\bar{x} + \bar{y}) \cdot (x + y + \bar{z})$$

25

3. Dạng chính tắc và dạng chuẩn của hàm Boole:

3.1. Các tích chuẩn (minterm) và tổng chuẩn (Maxterm):

- **Tích chuẩn (minterm):** m_i ($0 \leq i < 2^n - 1$) là các số hạng tích (AND) của n biến mà hàm Boole phụ thuộc với quy ước biến đó có bù nếu nó là 0 và không bù nếu là 1.
- **Tổng chuẩn (Maxterm):** M_i ($0 \leq i < 2^n - 1$) là các số hạng tổng (OR) của n biến mà hàm Boole phụ thuộc với quy ước biến đó có bù nếu nó là 1 và không bù nếu là 0.

x y z	minterm	Maxterm
0 0 0	$m_0 = \bar{x} \bar{y} \bar{z}$	$M_0 = x + y + z$
0 0 1	$m_1 = \bar{x} \bar{y} z$	$M_1 = x + y + \bar{z}$
0 1 0	$m_2 = \bar{x} y \bar{z}$	$M_2 = x + \bar{y} + z$
0 1 1	$m_3 = \bar{x} y z$	$M_3 = x + \bar{y} + \bar{z}$
1 0 0	$m_4 = x \bar{y} \bar{z}$	$M_4 = \bar{x} + y + z$
1 0 1	$m_5 = x \bar{y} z$	$M_5 = \bar{x} + y + \bar{z}$
1 1 0	$m_6 = x y \bar{z}$	$M_6 = \bar{x} + \bar{y} + z$
1 1 1	$m_7 = x y z$	$M_7 = \bar{x} + \bar{y} + \bar{z}$

$$m_i = \bar{M}_i$$

26

3.2. Dạng chính tắc (Canonical Form):

a. Dạng chính tắc 1: là dạng tổng của các tích chuẩn (minterm) làm cho hàm Boole có giá trị 1

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$\begin{aligned}
 F(x, y, z) &= \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}z + xy\bar{z} + xyz \\
 &= m_1 + m_2 + m_5 + m_6 + m_7 \\
 &= \Sigma m(1, 2, 5, 6, 7) \\
 &= \Sigma(1, 2, 5, 6, 7)
 \end{aligned}$$

$$\begin{aligned}
 F(x, y, z) &= (x + y + z)(x + \bar{y} + \bar{z})(\bar{x} + y + z) \\
 &= M_0 \cdot M_3 \cdot M_4 \\
 &= \Pi M(0, 3, 4) = \Pi(0, 3, 4)
 \end{aligned}$$

b. Dạng chính tắc 2: là dạng tích của các tổng chuẩn (Maxterm) làm cho hàm Boole có giá trị 0

27

* Trường hợp hàm Boole tùy định (don't care):

Hàm Boole n biến có thể không được định nghĩa hết tất cả 2^n tổ hợp của n biến phụ thuộc. Khi đó tại các tổ hợp không sử dụng này, hàm Boole sẽ nhận giá trị tùy định (don't care), nghĩa là hàm Boole có thể nhận giá trị 0 hoặc 1.

x	y	z	F
0	0	0	X
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	X

$$\begin{aligned}
 F(x, y, z) &= \Sigma(1, 2, 5, 6) + d(0, 7) \\
 &= \Pi(3, 4) \cdot D(0, 7)
 \end{aligned}$$

28

3.3. Dạng chuẩn (Standard Form):

a. Dạng chuẩn 1: là dạng tổng các tích (S.O.P – Sum of Product)

$$F(x, y, z) = xy + z$$

$$\begin{aligned} * F(x, y, z) &= xy + z \\ &= xy(\bar{z} + z) + (\bar{x} + x)(\bar{y} + y)z \\ &= xy\bar{z} + xyz + \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + \cancel{xyz} \\ &= m_6 + m_7 + m_1 + m_5 + m_3 \\ &= \underline{\Sigma(1, 3, 5, 6, 7)} \end{aligned}$$

$$\begin{aligned} * F(x, y, z) &= xy + z \\ &= (x + z)(y + z) \\ &= (x + \bar{y}y + z)(\bar{x}x + y + z) \\ &= (x + \bar{y} + z)(x + y + z)(\bar{x} + y + z) \cancel{(x + y + z)} \\ &= M_2 \cdot M_0 \cdot M_4 \\ &= \underline{\Pi(0, 2, 4)} \end{aligned}$$

29

b. Dạng chuẩn 2: là dạng tích các tổng (P.O.S – Product of Sum)

$$F(x, y, z) = (x + z')y'$$

$$\begin{aligned} * F(x, y, z) &= (x + \bar{z})\bar{y} = x\bar{y} + \bar{y}\bar{z} \\ &= x\bar{y}(\bar{z} + z) + (\bar{x} + x)\bar{y}\bar{z} \\ &= x\bar{y}\bar{z} + x\bar{y}z + \bar{x}\bar{y}\bar{z} + \cancel{x\bar{y}\bar{z}} \\ &= m_4 + m_5 + m_0 \\ &= \underline{\Sigma(0, 4, 5)} \end{aligned}$$

$$\begin{aligned} * F(x, y, z) &= (x + \bar{z})\bar{y} \\ &= (x + \bar{y}y + \bar{z})(\bar{x}x + \bar{y} + \bar{z}z) \\ &= (x + \bar{y} + \bar{z})(x + y + \bar{z}) \\ &\quad (\bar{x} + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z) \cancel{(x + \bar{y} + \bar{z})(x + \bar{y} + z)} \\ &= M_3 \cdot M_1 \cdot M_7 \cdot M_6 \cdot M_2 \\ &= \underline{\Pi(1, 2, 3, 6, 7)} \end{aligned}$$

30

Loại bỏ số hạng thừa

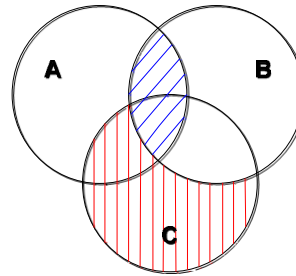
- Trong ví dụ sau, AC là số hạng thừa:

$$F(A,B,C) = AB + \bar{B}C + AC$$

Tối thiểu hóa?

Ta có:

$$\begin{aligned} F &= AB + \bar{B}C + AC(B + \bar{B}) \\ &= AB + \bar{B}C + ACB + AC\bar{B} \\ &= AB + ABC + \bar{B}C + A\bar{B}C \\ &= AB(1 + C) + \bar{B}C(1 + A) \\ &= AB + \bar{B}C \end{aligned}$$



31

III. RÚT GỌN HÀM BOOLE:

Rút gọn (tối thiểu hóa) hàm Boole nghĩa là đưa hàm Boole về dạng biểu diễn đơn giản nhất, sao cho:

- Biểu thức có chứa ít nhất các thừa số và mỗi thừa số chứa ít nhất các biến.
- Mạch logic thực hiện có chứa ít nhất các vi mạch số.

1. Phương pháp đại số:

Dùng các định lý và tiên đề để rút gọn hàm.

$$\begin{aligned} F(A, B, C) &= \Sigma(2, 3, 5, 6, 7) \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC \\ &= \bar{A}B(\bar{C} + C) + AC(\bar{B} + B) + AB(\bar{C} + C) \\ &= \bar{A}B + AC + AB \\ &= (\bar{A} + A)B + AC \\ &= B + AC \end{aligned}$$

32

VD:

1/ Tối giản biểu thức sau: $Z = \overline{(\overline{A} + C).(B + \overline{D})}$

2/ Dùng cổng NAND và cổng NOT để vẽ mạch điện có biểu thức

$$Z = \overline{A} + \overline{B} + C$$

3/ Dùng cổng AND, cổng OR và cổng NOT để vẽ mạch điện có biểu thức

(a) $X = \overline{AB(C + D)}$

(b) $Z = \overline{A + B + \overline{CDE}} + \overline{BCD}$

(c) $Y = \overline{M + N} + \overline{PQ}$

(d) $X = \overline{W + P\overline{Q}}$

(e) $Z = MN(P + \overline{N})$

33

4/ Dùng định luật De Morgan tối giản biểu thức:

(a) $Y = \overline{\overline{ABC}}$

(b) $Y = \overline{\overline{A} + \overline{BC}}$

(c) $Y = \overline{\overline{ABCD}}$

(d) $Y = \overline{\overline{A(B + \overline{C})D}}$

(e) $Y = \overline{(M + \overline{N})(\overline{M} + N)}$

(f) $Y = \overline{\overline{\overline{ABCD}}}$

34

2. Phương pháp bìa Các-nô (Karnaugh):

a. Cách biểu diễn:

- Bìa K gồm các ô vuông, mỗi ô vuông biểu diễn cho tổ hợp n biến. Như vậy bìa K cho n biến sẽ có 2^n ô.
- Hai ô được gọi là kề nhau khi tổ hợp biến mà chúng biểu diễn chỉ khác nhau 1 biến.
- Trong ô sẽ ghi giá trị tương ứng của hàm Boole tại tổ hợp đó. Ở dạng chính tắc 1 thì đưa các giá trị 1 và X lên các ô, không đưa các giá trị 0. Ngược lại, dạng chính tắc 2 thì chỉ đưa giá trị 0 và X.

* Bìa 2 biến:

$$F(A, B) = \Sigma(0, 2) + d(3) = \prod(1) \cdot D(3)$$

F	A		0	1
	B		0	1
0			0	2
1			1	3

F	A		0	1
	B		0	1
0			1	1
1				X

F	A		0	1
	B		0	1
0				
1			0	X

35

* Bìa 3 biến:

F	AB				
	C	00	01	11	10
0		0	2	6	4
1		1	3	7	5

$$F(A, B, C) = \Sigma(2, 4, 7) + d(0, 1) = \prod(3, 5, 6) \cdot D(0, 1)$$

F \ AB	C			
	00	01	11	10
0	X	1		1
1	X		1	

		AB			
		00	01	11	10
C	0	X		0	
	1	X	0		0

36

* Bìa 4 biến:

F \ AB	CD			
	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

* Bìa 5 biến:

F \ BC	A = 0				A = 1			
	00	01	11	10	10	11	01	00
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

37

b. Rút gọn bìa Karnaugh:

* Nguyên tắc:

- Liên kết đôi: Khi liên kết (OR) hai ô có giá trị 1 (Ô_1) kề cận với nhau trên bìa K, ta sẽ được 1 số hạng tích mất đi 1 biến so với tích chuẩn (biến mất đi là biến khác nhau giữa 2 ô). Hoặc khi liên kết (AND) hai ô có giá trị 0 (Ô_0) kề cận với nhau trên bìa K, ta sẽ được 1 số hạng tổng mất đi 1 biến so với tổng chuẩn (biến mất đi là biến khác nhau giữa 2 ô).

F \ AB	C			
	00	01	11	10
0		1	1	
1				

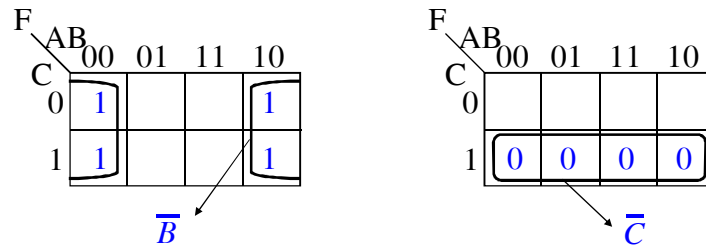
$B\bar{C}$

F \ AB	C			
	00	01	11	10
0				0
1				0

$\bar{A} + B$

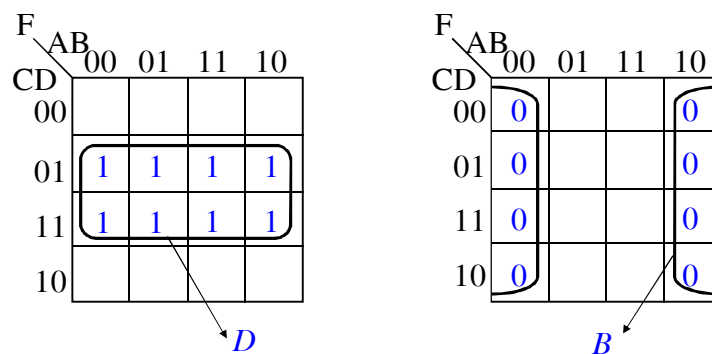
38

- Liên kết 4: Tương tự như liên kết đôi khi liên kết 4 Ô_1 hoặc 4 Ô_0 kề cận với nhau, ta sẽ loại đi được 2 biến (2 biến khác nhau giữa 4 ô)



39

- Liên kết 8: liên kết 8 ô kề cận với nhau, ta sẽ loại đi được 3 biến (3 biến khác nhau giữa 8 ô)



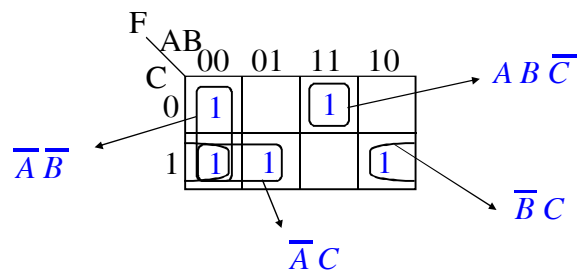
- Liên kết 2^k : khi ta liên kết 2^k Ô_1 hoặc 2^k Ô_0 kề cận với nhau ta sẽ loại đi được k biến (k biến khác nhau giữa 2^k ô)

40

* Các bước thực hiện rút gọn theo dạng S.O.P:

- ✓ Biểu diễn các Ô_1 lên bìa Karnaugh
- ✓ Thực hiện các liên kết có thể có sao cho các Ô_1 được liên kết ít nhất 1 lần; mỗi liên kết cho ta 1 số hạng tích. (Nếu Ô_1 không có kề cận với các Ô_1 khác thì ta có liên kết 1: số hạng tích chính bằng minterm của ô đó).
- ✓ Biểu thức rút gọn có được bằng cách lấy tổng (OR) của các số hạng tích liên kết trên.

$$F(A, B, C) = \Sigma(0, 1, 3, 5, 6) = \overline{A}\overline{B} + \overline{A}C + \overline{B}C + AB\overline{C}$$



41

* Ví dụ 1: $Y = \overline{A}\overline{B}C + A\overline{B}\overline{C} + \overline{A}B\overline{C}$

	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
\overline{C}	0	0	0	1
C	1	0	0	1

$$Y = A\overline{B} + \overline{B}C$$

* Ví dụ 2: $Y = \overline{A}\overline{B}\overline{C} + \overline{B}C + \overline{A}B$

	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
\overline{C}	1	1	0	0
C	1	1	0	1

$$Y = \overline{A} + \overline{B}C$$

42

* Ví dụ 3:

a) $Y = \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + ABCD + A\overline{B}CD + ABC\overline{D}$

	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}\overline{D}$	1			
$\overline{C}D$			1	1
CD			1	1
$C\overline{D}$			1	

$$Y = AD + ABC + \overline{A}\overline{B}\overline{C}\overline{D}$$

b) $Y = \overline{A}\overline{B}\overline{C} + \overline{A}BC + ABC + A\overline{B}\overline{C} + A\overline{B}C$

c) $Y = (\overline{C} + \overline{D}) + \overline{A}C\overline{D} + A\overline{B}\overline{C} + A\overline{B}CD + AC\overline{D}$

43

BT:

1) Cực tiểu hàm $f(A, B, C) = \sum^3 (0, 5, 6, 7)$

2) Cực tiểu hàm $f(A, B, C, D) = \sum^4 (1, 5, 6, 7, 10, 13, 15)$

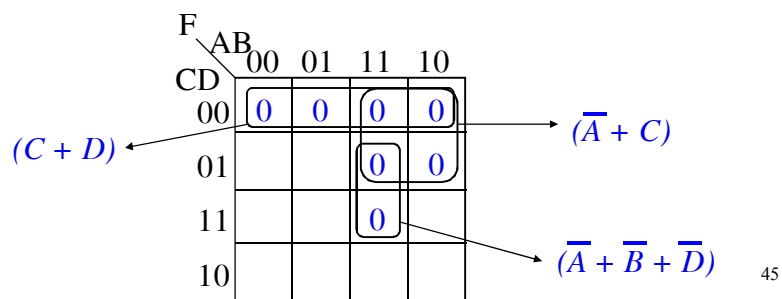
44

* Các bước thực hiện rút gọn theo dạng P.O.S:

- ✓ Biểu diễn các Ô_0 lên bìa Karnaugh
- ✓ Thực hiện các liên kết có thể có sao cho các Ô_0 được liên kết ít nhất 1 lần; mỗi liên kết cho ta 1 số hạng tổng.
- ✓ Biểu thức rút gọn có được bằng cách lấy tích (AND) của các số hạng tổng liên kết trên.

$$F(A, B, C, D) = \Pi(0, 4, 8, 9, 12, 13, 15)$$

$$= (C + D)(\bar{A} + C)(\bar{A} + \bar{B} + \bar{D})$$

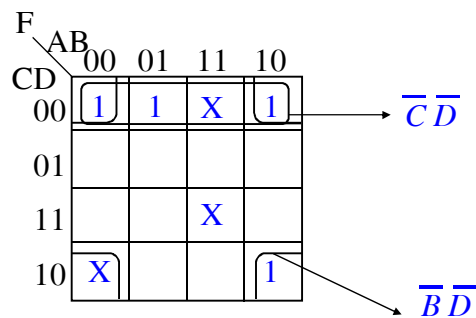


45

* Trường hợp rút gọn hàm Boole có tùy định: thì ta có thể coi các Ô tùy định này là Ô_1 hoặc Ô_0 sao cho có lợi khi liên kết (nghĩa là có được liên kết nhiều Ô kề cận nhất)

$$F(A, B, C, D) = \Sigma(0, 4, 8, 10) + d(2, 12, 15)$$

$$= \bar{B}\bar{D} + \bar{C}\bar{D}$$



46

$$F(A, B, C, D) = \Pi(0, 2, 3, 4, 6, 10, 14) \cdot D(8, 9, 11, 12, 13)$$

$$= D(B + \overline{C})$$

F		AB			
CD		00	01	11	10
	00	0	0	X	X
	01			X	X
	11	0			X
	10	0	0	0	0

→ D

→ (B + \overline{C})

47

* Chú ý:

- ✓ Ưu tiên liên kết cho các ô chỉ có 1 kiểu liên kết (phải là liên kết có nhiều ô nhất).
- ✓ Khi liên kết phải đảm bảo có chứa ít nhất 1 ô chưa được liên kết lần nào.
- ✓ Có thể có nhiều cách liên kết có kết quả tương đương nhau
- ✓ Ta coi các tùy định như là những ô đã liên kết rồi.

Vd: Rút gọn các hàm

$$F_1(A, B, C, D) = \Sigma(1, 3, 5, 12, 13, 14, 15) + d(7, 8, 9)$$

$$F_2(A, B, C, D) = \Pi(1, 3, 7, 11, 15) \cdot D(0, 2, 5)$$

$$F_1(A, B, C, D, E) = \Sigma(1, 3, 5, 7, 12, 14, 29, 31) + d(13, 15, 17, 19, 20, 21, 22, 23)$$

$$F_2(A, B, C, D, E) = \Pi(0, 8, 12, 13, 16, 18, 28, 30) \cdot D(2, 6, 10, 14, 15, 24, 26)$$

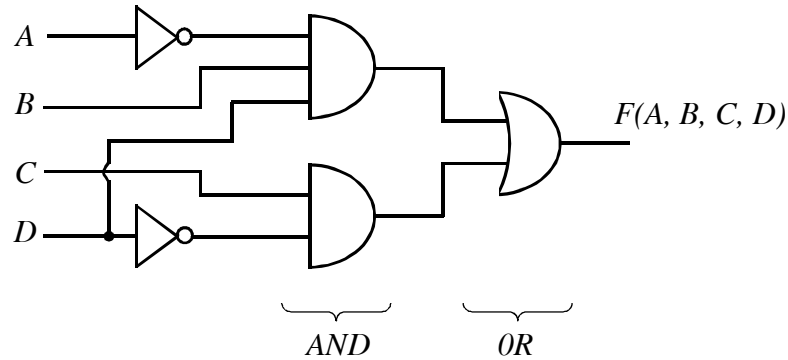
48

IV. THỰC HIỆN HÀM BOOLE BẰNG CỔNG LOGIC

1. Cấu trúc cổng AND_OR:

Cấu trúc AND_OR là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng tổng các tích (S.O.P)

$$F(A, B, C, D) = \bar{A} B D + C \bar{D}$$

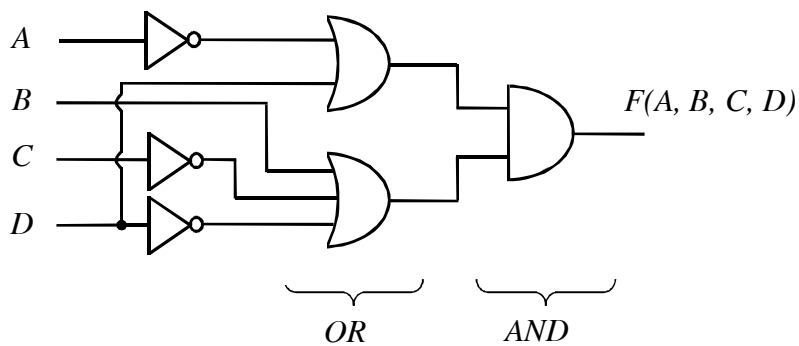


49

2. Cấu trúc cổng OR_AND :

Cấu trúc OR_AND là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng tích các tổng (P.O.S).

$$F(A, B, C, D) = (\bar{A} + D) (B + \bar{C} + \bar{D})$$



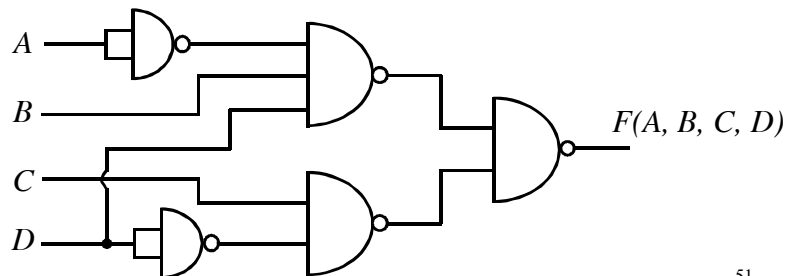
50

3. Cấu trúc toàn cổng NAND:

Cấu trúc NAND là sơ đồ logic thực hiện cho hàm Boole có biểu thức là dạng bù của 1 số hạng tích.

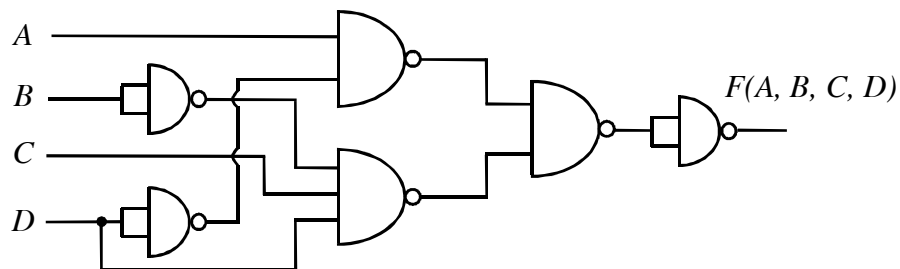
- Dùng định luật De-Morgan để biến đổi số hạng tổng thành tích.
- Cổng NOT cũng được thay thế bằng cổng NAND

$$\begin{aligned}F(A, B, C, D) &= \overline{\overline{\overline{A} B D} + \overline{C \overline{D}}} \\&= \overline{\overline{A} B D} \cdot \overline{\overline{C \overline{D}}}\end{aligned}$$



51

$$\begin{aligned}F(A, B, C, D) &= \overline{\overline{\overline{A} + D}} \cdot \overline{\overline{B + \overline{C} + \overline{D}}} \\&= \overline{\overline{A} D} \cdot \overline{\overline{B C D}}\end{aligned}$$

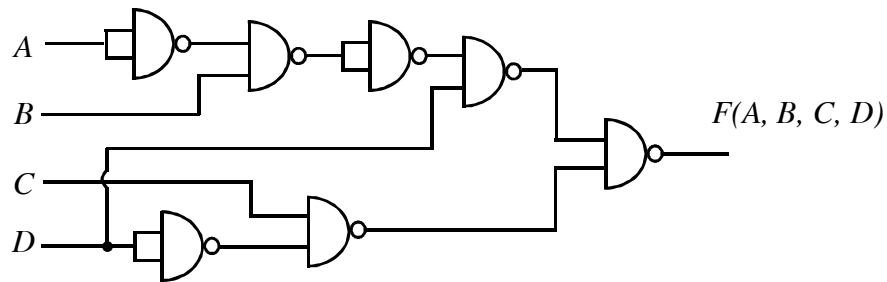


52

- Trong thực tế người ta chỉ sử dụng 1 loại cổng NAND 2 ngõ vào; khi đó ta phải biến đổi biểu thức sao cho chỉ có dạng bù trên 1 số hạng tích chỉ có 2 biến

$$F(A, B, C, D) = \overline{\overline{A} B D} \cdot \overline{C \overline{D}}$$

$$= \overline{\overline{A} B D} \cdot \overline{C \overline{D}}$$



53

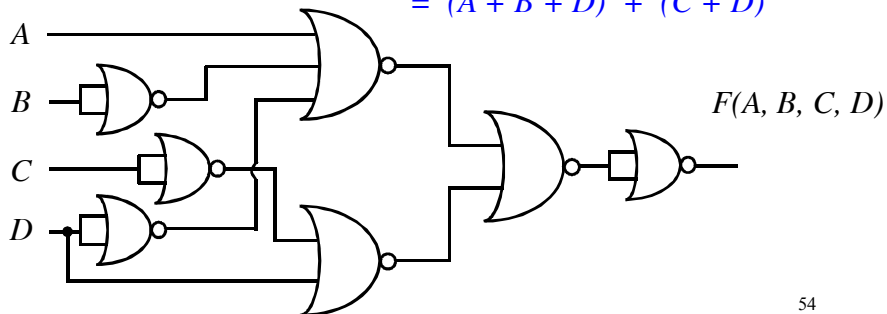
4. Cấu trúc toàn cổng NOR:

Cấu trúc NOR là sơ đồ logic thực hiện cho hàm Boole có biểu thức là dạng bù của 1 số hạng tổng.

- Dùng định lý De-Morgan để biến đổi số hạng tích thành tổng
- Cổng NOT cũng được thay thế bằng cổng NOR

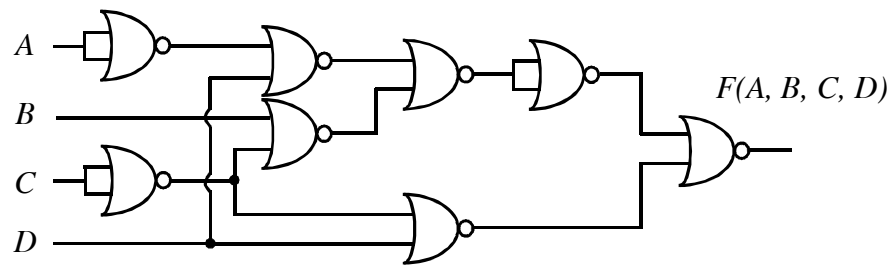
$$F(A, B, C, D) = \overline{\overline{A} B D} + \overline{C \overline{D}}$$

$$= \overline{(A + \overline{B} + \overline{D})} + \overline{(\overline{C} + D)}$$



54

$$\begin{aligned}
 F(A, B, C, D) &= \overline{\overline{(\bar{A} + D)(B + \bar{C})(\bar{C} + D)}} \\
 &= \overline{\overline{(\bar{A} + D)} + \overline{(B + \bar{C})} + \overline{(\bar{C} + D)}} \\
 &= \overline{(\bar{A} + D) + (B + \bar{C}) + (\bar{C} + D)}
 \end{aligned}$$



55